2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Real-time Filter for Human Pose Estimation based on Denoising Diffusion Models for Edge Devices

Chiara Bozzini*, Michele Boldo*, Enrico Martini, and Nicola Bombieri

***Abstract*— Human Pose Estimation (HPE) is increasingly utilized across various sectors, from healthcare to Industry 5.0. To address the inherent inaccuracies in CNN-based HPE systems, filtering models are commonly employed to refine and improve inference results. However, state-of-the-art filtering models often require substantial computational resources, limiting their applicability in resource-constrained environments. To overcome this limitation, we propose a real-time filtering approach based on denoising diffusion models (DM) specifically optimized for edge devices. Through a micro-benchmarking process, we analyze the DM adaptability to different types and levels of noise and determine the optimal setup for specific application scenarios. We present a real-time filter that takes advantage of the DM setup with two configurations to address different application scenarios. Using a widespread edge device, we evaluate the model's effectiveness in handling both synthetic and real noise generated by state-of-the-art HPE systems. The results demonstrate a significant improvement in real-time filtering performance with minimal computational overhead. The code is available on github.com/PARCO-LAB/LUT-DM-filters.**

***Index Terms*— Denoising Diffusion Models, Human pose estimation, Filtering, Edge Devices.**

I. Introduction

Human Pose Estimation (HPE) involves estimating geometric and kinetic data of the human body utilizing sensor-acquired data, particularly from videos. The advancements in neural network (NN) architectures, coupled with the fact that deep learning technology does not necessitate markers attached to the subject's body, have facilitated its broadening application across diverse domains. These include human-robot interaction, robotic task learning, and activity recognition [1]–[5]. In the robotic field, HPE is gaining an increasing interest as it forms the basis of cooperation activities between humans and robots, enabling the robot to accurately anticipate human movements [6], [7].

Although these NN-based models can be potentially applied in many scenarios, they often exhibit inaccuracies caused by challenging subject poses, occlusions, poor training data and intrinsic model and sensor inaccuracy.

To address these limitations, numerous filtering techniques have been proposed in recent years, ranging from those based on spatio-temporal data analysis to the more sophisticated methods based on bio-mechanical models [8]. A particularly promising category of filtering methodologies revolves around probabilistic denoising diffusion models (DDPMs), as outlined in the work of Ho et al. [9]. These NN-based models incorporate a denoising mechanism characterized by a series of iterative refinement steps. The iterative process aims to gradually improve the probabilistic distribution of the noisy input data, ultimately yielding high-quality results.

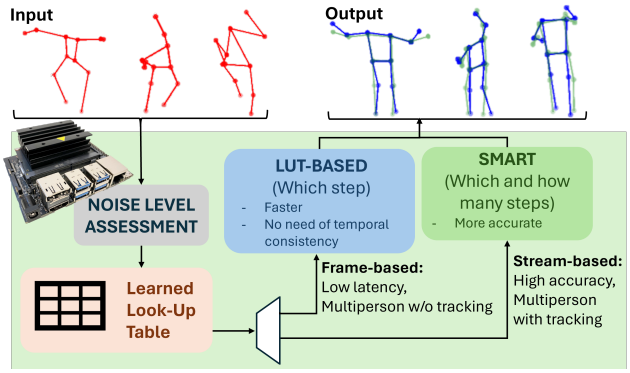

Fig. 1: Overview of the proposed real-time DM-based filtering technique

The standard implementation of DDPM requires significant computational resources and introduces considerable latency, thereby restricting its applicability in resource-constrained edge devices and scenarios with temporal constraints. In contexts such as robotics, where real-time processing and responsiveness are critical, these limitations become even more pronounced. To mitigate this issue, some approaches leverage Denoising Diffusion Implicit Models (DDIM) [10], to achieve high accuracy while employing a reduced number of refinement steps. However, the effectiveness of each individual step and the optimal number of consecutive steps heavily depend on the sample being denoised. An inappropriate selection of these two parameters may even result in a significant deterioration of the HPE prediction.

Recently, DDIM have been applied to implement 3D pose estimation and 2D-to-3D pose lifting without any constraints on computational resources [11], [12]. At present, there is no methodology proposed for using diffusion models for HPE refinement on *edge* devices with limited computational resources and in scenarios with temporal constraints. This work tackles this challenge by introducing a micro-benchmarking methodology to characterize the behaviour of the diffusion model under different levels of noise. We then propose a real-time filter with two configurations (see Fig. 1). At runtime,

This work has been supported by the "PREPARE" project (n. F/310130/05/X56 - CUP: B39J23001730005) - D.M. MiSE 31/12/2021, and the PNRR research activities of the consortium iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 –D.D. 1058 23/06/2022, ECS 00000043). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. Authors are with the Department of Engineering for Innovation Medicine, University of Verona, Italy - name.surname@univr.it
* equal contribution.

it selects the optimal DM refinement step thanks to the micro-benchmarking results in the first configuration or it dynamically selects the optimal sequence of refinement steps in the second configuration.

The main contributions of this work are as follows:

- Introducing a micro-benchmarking methodology to analyze how a diffusion model responds to inputs containing various levels of synthetic noise. This methodology also generates a lookup table (LUT) for the selection of the corresponding optimal diffusion model refinement steps.
- Proposing a zero-latency, real-time diffusion model-based filter that, given a noisy sample at the time, applies the refinement step identified through the lookup table to filter the sample.
- Introducing a zero-latency, real-time diffusion model-based *dynamic* filter that, given a noisy sample at a time and utilizing the history of only one prediction and the corresponding noise, selects the optimal number and diffusion steps to filter the sample.
- Conducting an extensive evaluation of the proposed solutions on two standard and large datasets, employing both synthetic noise and state-of-the-art 3D HPE techniques.

The code is available on github.com/PARCO-LAB/LUT-DM-filters.

II. Background

Several HPE solutions have demonstrated impressive accuracy in estimating the pose of individual subjects from images and videos, leveraging both 2D and 3D pose annotations. However, these solutions exhibit accuracy constraints, particularly in scenarios characterized by high occlusion, the presence of multiple individuals, and non-standard poses [13], [14].

Different filtering techniques have been proposed to address this accuracy limitation [8]. Traditional filters such as Gaussian and Savitzky-Golay are used for motion refinement, alongside simple moving average and exponential moving average filters [15]. State observers, including Kalman filters (KF), are employed to smooth measurements based on positions, velocities, or accelerations [16]. Learned approaches, such as convolutional and bidirectional recurrent autoencoders, provide compressed representations for error removal and have demonstrated effective filtering performance [17], [18]. The latest advancements in HPE and filtering techniques leverage Denoising Diffusion Probabilistic Models (DDPM) [9]. These models belong to a class of generative models that exploit diffusion processes to denoise data and generate samples by iteratively refining an input distribution corrupted by noise. DDPM exhibit exceptional performance across various tasks, including image generation [19] and editing [20]. Moreover, their applications extend to 3D reconstruction [21], image inpainting [22], and human motion generation [23]. DDPM, however, present significant computational requirements. Consequently, efforts have been made to simplify them, resulting in the development of Denoising Diffusion Implicit Models (DDIM) [10]. DDIM utilize non-Markovian diffusion processes to expedite sampling, thereby aiming to mitigate computational overhead significantly.

Over the past two years, there has been a notable increase in research employing diffusion models to enhance human pose prediction accuracy. Holmquist et al. [11] introduced an innovative pose estimation framework that integrates a diffusion model, treating 3D pose estimation as a reverse diffusion process. In [24], the authors present a monocular 2D-to-3D methodology utilizing Graph Convolutional Networks (GCNs) as denoising functions within diffusion models. This approach facilitates the explicit learning of the interconnections among human joints within 3D space.

In [25], the authors propose a diffusion-based approach to 3D HPE, introducing joint-wise reprojection-based multi-hypothesis aggregation to improve accuracy, departing from traditional pose-level methods like averaging. Jiang et al. [26] used a truncated diffusion model that requires only the 3D pose and time step as input, eliminating the need for 2D input unlike previous models.

At the state of the art, neither of these methods investigates the applicability of diffusion models in environments with computational and real-time constraints. Our work propose two different real-time filtering strategies based on a single and an optimal number of steps of a diffusion model. The proposed analysis focuses on the DM adaptability to different types and levels of noise, with the aim of determining the optimal setup for specific application scenarios.

III. Diffusion model setup and use in the real-time filtering

Given a pre-trained diffusion model (DM), the idea is to characterize the behaviour of this generative AI model in the denoising process of a human pose estimation platform. To achieve this, we propose a *micro-benchmarking* methodology whereby various types and intensities of noise are injected into each HPE sample of a dataset. For each injected sample, we measure the denoising efficacy of each DM step. The result of this offline analysis is a look up table, which serves as input for the proposed real-time filter.

A. DM micro-benchmarking methodology

Given a diffusion model DM trained with N diffusion steps, the micro-benchmarking phase aims at systematically assessing the DM efficacy in denoising different types of noise. Fig. 2 shows the methodology overview. Starting from a HPE sample, kp, and a set of noise instances, $NOISE$, where each instance is a tuple $< t, s >$ (type and intensity of noise), we generate the set of noisy HPE samples KP_n, one sample per tuple.

We then apply the DM to denoise each sample $kp_n \in KP_n$ by running each DM step of the diffusion process individually (i.e., [[0], ..., $[N-1]$, $[N]$]. The result is the set of filtered samples $KP_f = \{kp_f^0, ..., kp_f^i, ..., kp_f^N\}$ where i is the diffusion step. We extrapolate the *mean per joint position error* $MPJPE_{KP_f}$ for each filtered sample, kp_f^i:

$$mpjpe(kp_f^i, kp) = \frac{1}{j}\sum_{t=1}^{j} \|\mathbf{p}_{kp_f^i}^t - \mathbf{p}_{kp}^t\| \quad (1)$$

where j is the number of body joints, and p the 3D position of the joint. We undertake this process since each step effectively removes a variable degree of noise from kp_n.

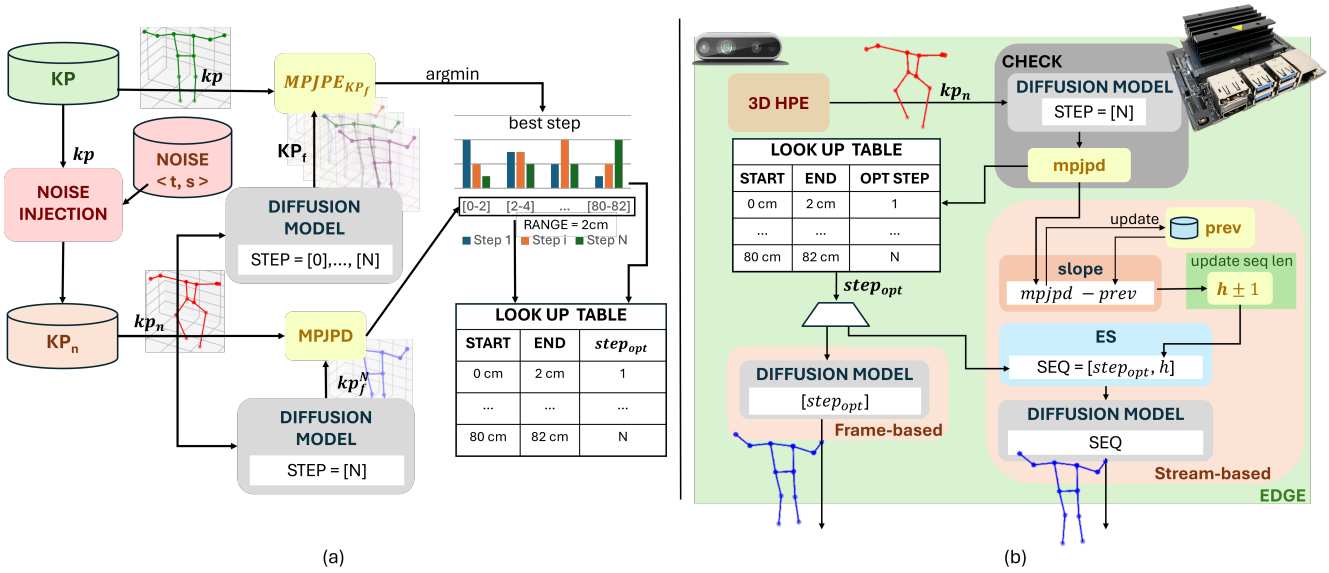

Fig. 2: DM micro-benchmarking methodology (a), and LUT-based real-time filtering (b)

Fig. 3 shows, as an example, the impact of each step of a DM with 51 diffusion steps, represented as the mean of $MPJPE_{KP_f}$ of each sample kp_n of the Human3.6M [27] dataset, in refining four different levels of noise. In cases where the *signal-to-noise ratio* (SNR) is lower, indicating a higher level of added noise, the final DM step, which yields the lowest $MPJPE_{KP_f}$, becomes the most effective. Conversely, as the SNR increases, indicating a less noisy input, the optimal refinement step tends to shift towards the earlier steps in the process. Then, for each kp_f^N, we calculate the mean per joint position Euclidean distance $mpjpd$ to kp_n as follows:

$$mpjpd(kp_f^N, kp_n) = \frac{1}{j}\sum_{t=1}^{j} \|\mathbf{p}^t_{kp_f^N} - \mathbf{p}^t_{kp_n}\| \qquad (2)$$

where j is the number of body joints, and p the 3D position of the joint. It is important to note that we calculate the distance by considering the joint positions filtered with the most aggressive step of the DM (i.e., N). We collect the set of these distances for each kp of the dataset, $MPJPD$.

Given a sample kp_n and the corresponding $mpjpd(kp_n, kp_f^N)$ measured in real-time, the idea is to take advantage of the correlation between $mpjpd(kp_n, kp_f^N)$ and $mpjpe(kp_n, kp)$ to select the most efficient DM step to denoise kp_n. Fig. 4 shows, for example, the correlation between $MPJPD$ and $MPJPE_{KP_n} = \{mpjpe(kp_n, kp) \mid \forall kp_n \in KP_n\}$ calculated in the Human3.6M [27] with different levels of Gaussian noise injected in the samples.

We determine the optimal DM step, $step_{opt}$, for a given $mpjpd$ by finding the value of i that corresponds to the minimum value in $\{MPJPE_{kp_f^i} \mid \forall i = 0, \ldots, N-1, N\}$ (i.e. argmin function). The result is a set of tuples $\{< mpjpd_{kp_f^N}, step_{opt} >, \forall kp_n \in KP_n\}$.

After grouping this set into bins, based on $mpjpd_{kp_f^N}$ with a specific range r, we iterate over each group, extracting the mode step of the group, which corresponds to the step that appears most frequently within the considered range.

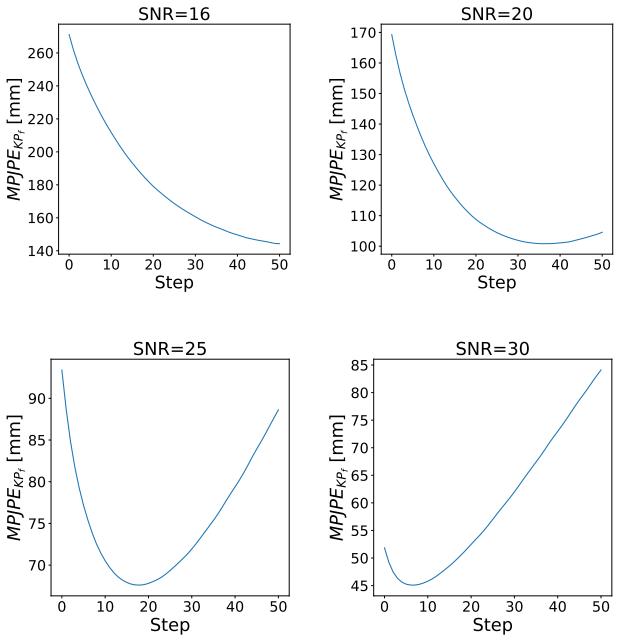

Fig. 3: Analysis of $MPJPE_{KP_f}$ variations across an entire Human3.6m dataset under different step application and levels of injected additive gaussian noise, by varying the signal-to-noise ration (SNR). As the SNR is higher, less noise is injected.

The result is a look-up table LUT=$[start, end, step_{opt}]$, where $start$ and end denote the start and end of the range under consideration, and $step_{opt}$ is the mode step for the considered range.

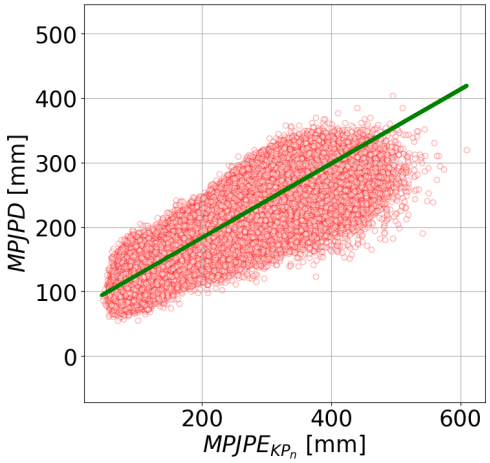

Fig. 4: Correlation between $MPJPD$ and $MPJPE_{KP_n}$

B. LUT-based real-time filtering

The objective is to minimize the number of DM steps required to refine noisy samples while meeting real-time constraints. Drawing on empirical evaluations from [12] and [10], which suggest that the efficiency of the diffusion model is not linearly correlated with the number of refinement steps applied to the noisy sample, our approach entails selecting the optimal number of refinement steps for each noisy sample.

We propose two possible configurations (see Fig. 2(b)): A DM-based filter called *LUT-DM* (Frame-based in the figure) that receives one noisy sample kp_n at the time and the LUT generated during the micro-benchmarking phase to filter kp_n; and a DM-based filter called *Smart-DM* (Stream-based in the figure) that accepts the current noisy sample kp_n and the previous noisy sample as input, along with the LUT to filter kp_n.

1) ***LUT-DM****:* This filter first applies the N-th DM step to obtain kp_f^N and then calculates the $mpjpd(kp_f^N, kp_n)$ value. It extrapolates the optimal step through the LUT.

Thanks to the two-steps computation (i.e., $mpjpd$ calculation and refinement), this approach does not require temporal consistency across samples and guarantees low-latency. Temporal consistency is crucial in multi-person scenarios where the identification number of each sample kp, which is mandatory to extrapolate the mean per joint errors and distance may be inconsistent between frames.

2) ***Smart-DM****:* This real-time filter is an evolution of *LUT-DM* for scenarios where multiple refinement steps are feasible. Algorithm 1 shows the pseudocode. It starts from the HPE sample kp_n, the output of the first diffusion step kp_f^N, the $mpjpd$ of the previous sample (i.e. $prev$), the number of diffusion steps h, and the LUT. First, it computes the current $mpjpd$ and through the LUT, it extrapolates the optimal diffusion step $step_{opt}$. If this is the first iteration of the algorithm (i.e. the $prev$ is empty), the algorithm returns $step_{opt}$. If there exists a $prev$, the algorithm computes the slope of the line which passes through the two points, the current $mpjpd$ and $prev$, respectively. If the slope is greater than 0, there is an increase in the $mpjpd$ (i.e., the current input is worse than the previous). In this case, given that more refining steps remove more noise, it increases the number of steps h. Conversely, a negative $slope$ means that the current input is better than the previous. In this case, it decreases h. This concept shares similarities with a derivative controller, as the derivative of the error provides an immediate indication of the error's rate of change over time. This allows the system to anticipate and rectify deviations in the error. Finally, the algorithm computes the optimal step list as follows:

$$ES(step_{opt}, h) = \{0, \frac{step_{opt}}{h-1}, 2\frac{step_{opt}}{h-1}, \ldots, step_{opt}\} \quad (3)$$

Such an *Equally Spaced* list consists of n steps, evenly distributed from the initial step 0 to the final step $step_{opt}$. As the refinement process involves reverse diffusion, these steps are applied in reverse order.

Algorithm 1 Smart-DM

Require: kp_n, kp_f^N, $prev$, h, LUT
1: $mpjpd \leftarrow$ Euclidean_distance(kp_n, kp_f^N)
2: $step_{opt} \leftarrow$ LUT$(mpjpd)$
3: $seq \leftarrow [step_{opt}]$
4: **if** $prev \neq \emptyset$ **then**
5: $\quad slope \leftarrow mpjpd - prev$
6: $\quad$ **if** $slope > 0$ **then**
7: $\quad\quad h \leftarrow h + 1$
8: $\quad$ **else**
9: $\quad\quad h \leftarrow h - 1$
10: $\quad$ **end if**
11: $\quad seq \leftarrow ES(step_{opt}, h)$
12: **end if**
13: $prev \leftarrow mpjpd$
14: **return** h, seq, $prev$

IV. Results

For the experimental evaluation, we started from the DM implementation proposed in [12]. We customized such a model in order to make it capable of handling a set of 12 human keypoints (i.e., knees, ankles, hips, shoulders, elbows, and wrists left and right). We trained the model on the ground truth of the *Human3.6M* [27] *training set*. We run the experimental evaluation of the proposed DM-based filters on a NVIDIA Jetson Xavier NX. This is a heterogeneous board equipped with a GPU accelerator with 384 CUDA cores and a 6-core processor, with a total of 8 GB of unified memory.

For the micro-benchmarking phase, we considered the same *Human3.6M* [27] as the reference dataset (KP). We injected additive white gaussian noise on the ground truth of the *train subjects*, by varying the signal-to-noise ratio (SNR) across levels 16, 20, 25, and 30. The result is an injected dataset (KP_n) of samples with average mpjpes of 287.9, 181.6, 102.2, and 57.4 mm, respectively. We also designed a scenario incorporating mixed SNR levels to introduce greater variability in the error distribution, aiming to simulate conditions more closely resembling real-world scenarios. We also evaluated the filters on the poses estimated

by a state-of-the-art HPE (i.e. *Openpose* [13]) from the videos of the *Human3.6M*. In addition, we performed a cross-dataset evaluation on the ground truth of *Total Capture* [28], injecting additive white gaussian noise. All datasets guarantee the temporal consistency of the HPE samples across the video frames.

By following the proposed micro-benchmarking methodology, we extrapolated the LUT. We implemented and compared the following diffusion-based filtering techniques:

- **DDPM** (51 steps): a standard DDPM implementation, which filters the noisy input by employing all feasible DM steps, as suggested in [12]. Despite its computational demands and inability to ensure real-time denoising on edge devices, we include it for comparative analysis of filtering effectiveness.
- **DDIM-2 steps**: The diffusion implicit model proposed in [12], which applies a static subset of DM steps. To ensure real-time performance (i.e., $\geq$ 10 FPS), we constrained the step number to two and selected the optimal steps, as recommended by the authors (i.e., [0,6]).
- **Oracle** (1 step): An offline DM filter that executes all possible steps individually and, by comparing the output to the ground truth, selects the pose which best minimizes the $MPJPE_{KP_f}$. This approach is unfeasible in real-world domains where real-time ground truth data is unavailable.
- **LUT-DM** (Ours, 1 step): The LUT-based filter proposed in Section III-B.1.
- **Smart-DM** (Ours, $\leq$ 5 steps): The dynamic LUT-based filter proposed in Section III-B.2 in which, to guarantee real-time performance on the adopted edge device ($\geq$ 10 FPS), we fixed the maximum step number to 5.

Table I presents the comparison of the different filtering techniques in terms of accuracy.

A. Results on the test set of Human3.6M

We run the first evaluation on the test set of Human3.6M with different and mixed SNR levels. We found that DDPM achieves the highest efficacy in denoising highly and moderately noisy inputs (i.e., SNR = 20 and SNR = 25). However, this model tends to over-filter the input in lower noisy scenarios (e.g., SNR = 30). This leads to a 5.7% worsening, in average, of the pose estimation accuracy. It is important to note that the DDPM filter executes 51 refining steps, rendering it impractical for real-time applications on edge devices. The DDIM filter, with only two steps, outperforms the DDPM in scenarios with lower noisy inputs. Our *LUT-DM* filter attains superior results in highly noisy scenarios, exhibiting an average accuracy improvement of 49.9%. In other scenarios, the *LUT-DM* filter achieves accuracy results comparable to the best performance observed between DDPM and DDIM. The proposed Smart filter consistently achieves the best results among all filters across all scenarios. However, as for DDPM and DDIM, it is not applicable when the temporal consistency of the HPE poses across video frames is not assured. In such instances, only the *LUT-DM* filter can be employed effectively.

B. Results with state-of-the-art 3D HPE

We also evaluated the efficacy of the different solutions in filtering one of the most efficient and widespread 3D HPE at the state-of-the-art. We triangulated the 2D OpenPose [13] results obtained on the videos of Human3.6M taken from different RGB cameras to obtain the 3D HPE dataset.

In this scenario, DDPM demonstrates the most effective filtering performance, yielding an average improvement of 19.6%. While DDIM and both the proposed filters also exhibit good results, their efficacy is slightly lower than DDPM. However, unlike DDPM, they operate in real-time (refer to the subsequent section for computational performance analysis).

It is important to note that, in this scenario, *LUT-DM* achieves superior results compared to DDIM, despite implementing only a single step. This underlines the significance of selecting the right step, as it evidently has a more substantial impact on accuracy than the mere implementation of multiple random or static DM steps.

C. Model Generalizability

We performed a cross-dataset evaluation of the filter efficiency. We used the DM model trained on *Human3.6M*, the Look-Up table extrapolated from *Human3.6M* and tested the filters on the *Total Capture* dataset [28]. Additive white gaussian noise was introduced into the data to achieve a mean of 181.7 mm and a standard deviation of 31.3 mm. Remarkably, even under these conditions, the *Smart* filter demonstrates denoising efficiency on par with that of DDPM. Furthermore, the *LUT-DM* filter significantly improves data quality, enhancing it by 29.7%

D. Real-time performance analysis

Table II presents a comparison of the different filtering techniques in terms of real-time performance. For reference, the column *One step* represents the performance metrics of a hypothetical methodology which executes only one refinement step. As expected, DDPM emerges as the most computationally demanding filter, executing 51 steps consistently, with an average latency of 1.63 seconds per sample (equivalent to 0.61 frames per second - FPS). In contrast, the standard DDIM model, which performs 2 DM steps, achieves a working frequency of 14.02 FPS, representing a speedup of ≈23x with respect to DDPM. Despite its lighter computational load, DDIM does not yield significant accuracy improvements. The *Smart-DM* filter, on average, executes 2.99 refinement steps along with an additional step for noise, resulting in a working frequency of 7.63 FPS, which signifies a speedup of 16.4x compared to DDPM. Lastly, *LUT-DM*, by executing a single DM step, stands out as the lightest filter, achieving a working frequency of 14.7 FPS, inclusive of the noise assessment step, with a speedup of 24.1x compared to DDPM.

V. Conclusions

We investigated the adoption of DDPM to refine real-time HPE data on resource-constrained edge devices. Through micro-benchmarking, we analyzed the model's responsiveness to various levels and types of noise. We developed a real-time filter with two configurations: *LUT-DM* and *Smart-DM*. Both configurations exhibit significantly higher refinement efficiency compared to state-of-the-art DM-based

TABLE I: Accuracy results

Dataset (Noise)	IN MPJPE (mm)	%	Oracle MPJPE (mm)	%	DDPM [12] MPJPE (mm)	%	DDIM [12] MPJPE (mm)	%	LUT-DM MPJPE (mm)	%	SMART-DM MPJPE (mm)	%
H36M - (SNR=16)	287.9 ± 49.6	-	142.4 ± 46.5	50.5%	154.3 ± 51.4	46.4%	235.4 ± 49.1	18.2%	144.3 ± 46.3	49.9%	145.6 ± 49.1	49.4%
H36M - (SNR=20)	181.6 ± 31.2	-	96.1 ± 25.5	47.1%	98.2 ± 27.5	45.9%	143.1 ± 31.3	21.2%	104.6 ± 26.1	42.4%	99.2 ± 26.9	45.4%
H36M - (SNR=25)	102.2 ± 17.6	-	64.6 ± 14.2	36.7%	70.3 ± 16.3	31.2%	77.6 ± 17.0	24.1%	73.6 ± 20.0	28.0%	71.8 ± 17.4	29.8%
H36M - (SNR=30)	57.4 ± 9.9	-	43.4 ± 8.5	24.4%	60.7 ± 13.1	-5.7%	45.1 ± 9.2	21.4%	46.7 ± 13.1	18.7%	46.0 ± 11.6	19.9%
H36M (Mixed SNR)	157.3 ± 92.9	-	86.7 ± 46.5	44.9%	95.9 ± 47.8	39.0%	125.3 ± 78.9	20.3%	92.3 ± 46.6	41.3%	90.6 ± 47.5	42.4%
3D HPE	148.7 ± 95.5	-	117.1 ± 88.90	21.3%	119.6 ± 85.9	19.6%	131.1 ± 92.13	11.9%	125.3 ± 88.9	15.8%	124.9 ± 86.1	16.0%
Total Capture	181.7 ± 31.3	-	115.6 ± 25.4	36.4%	119.9 ± 26.0	34.0%	149.4 ± 30.8	17.8%	127.6 ± 25.8	29.7%	122.5 ± 25.6	32.6%

TABLE II: Real-time performance results. The latency and FPS measures of *LUT-DM* and *Smart-DM* include the assessment step.

	One step	DDPM [12]	DDIM [12]	SMART	LUT-DM
Refinement steps	1	51	2	2.99	1
Assessment steps	-	-	-	1	1
Computation time (s)	0.034	1.63	0.07	0.131	0.068
FPS	29.4	0.61	14.02	7.63	14.7
Speedup w.r.t. DDPM	47.94x	-	22.98x	12.5x	24.1x

solutions in real-time scenarios, such as DDIM. The *LUT-DM* filter demonstrates the best real-time performance and remains applicable even when temporal consistency across samples is not guaranteed. On the other hand, *Smart-DM* achieves the highest filtering efficiency across all scenarios with guaranteed temporal consistency across samples.

REFERENCES

[1] A. A. Goksoy, S. An, and U. Y. Ogras, "Energy-efficient on-chip training for customized home-based rehabilitation systems," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, Jul. 2023.

[2] H. M. Clever, A. Kapusta, D. Park, Z. Erickson, Y. Chitalia, and C. C. Kemp, "3d human pose estimation on a configurable bed from a pressure image," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 54–61.

[3] A. Martínez-González, M. Villamizar, O. Canévet, and J.-M. Odobez, "Residual pose: A decoupled approach for depth-based 3d human pose estimation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 313–10 318.

[4] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, "3d human pose estimation in rgbd images for robotic task learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1986–1992.

[5] B. Reily, Q. Zhu, C. Reardon, and H. Zhang, "Simultaneous learning from human pose and object cues for real-time activity recognition," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8006–8012.

[6] L.-Y. Gui, K. Zhang, Y.-X. Wang, X. Liang, J. M. F. Moura, and M. Veloso, "Teaching robots to predict human motion," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 562–567.

[7] Z. Erickson, H. M. Clever, V. Gangaram, E. Xing, G. Turk, C. K. Liu, and C. C. Kemp, "Characterizing multidimensional capacitive servoing for physical human–robot interaction," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 357–372, 2023.

[8] E. Martini, A. Calanca, and N. Bombieri, "Denoising and completion filters for human motion software: a survey with code," *TechRxiv*, 2023.

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 2020-December, 2020.

[10] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *ICLR 2021 - 9th International Conference on Learning Representations*, 2021.

[11] K. Holmquist and B. Wandt, "Diffpose: Multi-hypothesis human pose estimation using diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 15 977–15 987.

[12] J. Gong, L. G. Foo, Z. Fan, Q. Ke, H. Rahmani, and J. Liu, "Diffpose: Toward more reliable 3d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[13] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 2016, pp. 483–499.

[15] M. Memar Ardestani and H. Yan, "Noise reduction in human motion-captured signals for computer animation based on b-spline filtering," *Sensors*, vol. 22, no. 12, p. 4629, 2022.

[16] Q. Wu and P. Boulanger, "Real-time estimation of missing markers for reconstruction of human motion," in *2011 XIII Symposium on Virtual Reality*. IEEE, 2011, pp. 161–168.

[17] P. Skurowski and M. Pawlyta, "Gap Reconstruction in Optical Motion Capture Sequences Using Neural Networks," *Sensors*, vol. 21, no. 18, p. 6115, sep 2021.

[18] A. Zeng, L. Yang, X. Ju, J. Li, J. Wang, and Q. Xu, "Smoothnet: A plug-and-play network for refining human poses in videos," in *European Conference on Computer Vision*. Springer, 2022.

[19] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8821–8831.

[20] S. Cao, W. Chai, S. Hao, Y. Zhang, H. Chen, and G. Wang, "Diffashion: Reference-based fashion design with structure-aware transfer by diffusion models," *IEEE Transactions on Multimedia*, vol. 26, pp. 3962–3975, 2024.

[21] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," 2022.

[22] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 461–11 471.

[23] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," 2022.

[24] J. Choi, D. Shim, and H. J. Kim, "Diffupose: Monocular 3d human pose estimation via denoising diffusion probabilistic model," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3773–3780.

[25] W. Shan, Z. Liu, X. Zhang, Z. Wang, K. Han, S. Wang, S. Ma, and W. Gao, "Diffusion-based 3d human pose estimation with multi-hypothesis aggregation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 14 761–14 771.

[26] Z. Jiang, Z. Zhou, L. Li, W. Chai, C.-Y. Yang, and J.-N. Hwang, "Back to optimization: Diffusion-based zero-shot 3d human pose estimation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2024, pp. 6142–6152.

[27] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[28] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, "Total capture: 3d human pose estimation fusing video and inertial sensors," in *2017 British Machine Vision Conference (BMVC)*, 2017.